Introduction au TALN (1)
Corpus et expressions rationnelles

3 janvier 2011

La version la plus récente de ce support de cours est probablement disponible à l'adresse : `http://www.fabienpoulard.info/download//Enseignements/EPUN/2011/`.

Exercice 1.1 (Sans machine – Compréhension des expressions rationnelles)

Pour chacune des expressions rationnelles (ou expressions régulières) suivantes, donner :

1. un exemple de chaîne reconnue ;
2. un exemple (pertinent) de chaîne non reconnue ;
3. la signification de l'expression.

Remarque : le symbole "␣" représente un espace.

1. `(␣*|\t*)`
2. `[␣\t]*`
3. `s/\bJeff\b/Jeff/i;`
4. `s/(\.\d\d[1-9]?)\d*/$1/;`
5. `s/(\.\d\d[1-9]?)\d+/$1/;`
6. `s/^From:(\S+)␣\(([^()]*)\)/Expéditeur:␣$1\nNom:␣$2\n/;`
7. `m/([a-z]{3,})[a-z]*\1/`
8. `s/␣+/␣/g;`

Exercice 1.2 (Sur machine – *grep* et *sed*)

Nous allons mettre en œuvre les connaissances acquises dans l'exercice précédent dans un contexte d'administration système. Pour ce faire, nous allons utiliser les outils *grep* et *sed*. L'objectif n'est pas d'utiliser les capacités particulières de ces outils, mais uniquement leur moteur interne de traitement des expressions rationnelles.
Vous vous contenterez de ces quelques éléments de syntaxe pour les deux outils :

– `grep --only-matching --extended-regexp` $< expression >$ $< Fichier >$ ou plus court `grep -o -E` $< expression >$ $< Fichier >$;
– l'option `-i` indique à *grep* de ne pas tenir compte de la casse des caractères ;
– `sed 's/expression/motif de remplacement/g'`

Vous trouverez les fichiers à analyser à l'adresse : `http://www.fabienpoulard.info/download/Enseignements/EPUN/2011/`.
Utilisez les expressions rationnelles pour :

1. lister les différentes adresses IP qui ont contacté le service *httpd* (*httpd-access.log*) ;
2. identifier les méthodes de connexion au serveur (*auth-access.log*) ;
3. les utilisateurs de *sudo* (*auth-access.log*) ;
4. les commandes lancées par *sudo* (*auth-access.log*) ;
5. les attaques par *ssh* et leurs origines (*auth-access.log*) ;
6. les clients qui provoquent des erreurs sur *httpd* (*httpd-error.log*).

Exercice 1.3 (Sur machine – Python et les expressions rationnelles)

Python offre un moteur de traitement des expressions rationnelles. Il suffit d'importer le module *re* (`import re`). La documentation de ce module est disponible à l'adresse : `http://docs.python.org/library/re.html`.

Voici un extrait de code Python permettant d'identifier les IP sources des attaques sur *ssh* et les dénombrer :

```
# Chargement du module pour les expressions rationnelles
import re
# Préparation de l'expression régulière
reg_attack = re.compile(".*Invalid user (\w+) from ([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}).*")
# Initialisation du dictionnaire pour compter les IP
dic_ip = {}
# Ouverture du fichier
fh = open("auth-access.log", "r")
for line in fh.readlines():
    # La ligne contient-elle l'information recherchée ?
    m = reg_attack.match(line)
    if not m is None:
        # Affichage du nom et de l'IP
        nom = m.groups()[0]
        ip  = m.groups()[1]
        print "Attaque depuis %s (login : %s)" % (ip,nom)
        # Ajout de l'IP au compteur
        if not dic_ip.has_key(ip):
            dic_ip[ip] = 1
        else:
            dic_ip[ip] += 1
fh.close()
```

Vous trouverez une version téléchargeable de ce script à l'adresse : `http://www.fabienpoulard.info/download/Enseignements/EPUN/2011/count-ip-ssh-attacks.py`.

Inspirez-vous de ce code pour extraire les données de l'exercice précédent et éventuellement opérer des traitements statistiques sur celles-ci.

Exercice 1.4 (Sur machine – Tokenisation en mots)

Vous trouverez à l'adresse ci-dessous une compilation des discours de Nicolas Sarkozy prononcés depuis le début de son mandat présidentiel :
`http://www.fabienpoulard.info/download//Enseignements/EPUN/2011/discours-sarkozy-all.tar.bz2`
Nous utiliserons ce corpus comme base de travail pour la suite de nos exercices.
Cet exercice se veut une petite introduction au découpage en mots et à l'utilisation des corpus.

1. La méthode `split` des objets expressions rationnels (ceux construits avec `re.compile`) permet de découper un texte en séparant les séquences par les motifs décrits par l'expression rationnelle. Essayez de découper le texte du corpus avec l'expression `[ ]`. Combien comptez-vous de mots ?
2. Proposez des améliorations pour ce motif.
3. Une autre approche du découpage en mots est de repérer, non plus ce qui sépare les mots, mais ce qui est un mot. La méthode `finditer` permet de parcourir les éléments reconnus. Essayer de découper le corpus en mots en utilisant l'expression `[a-z]+`. Combien comptez-vous de mots ?
4. Proposez des améliorations pour ce motif.

Exemple de code pour le chargement et le parcours des fichiers du corpus :

```
# Chargement des modules pour l'encodage et les expressions rationnelles
import re
import codecs
import os
# Préparation de l'expression rationnelle
reg_split = re.compile("[ \t]")
# Initialisation du compteur de mots
nbmots = 0
# Initialisation du dictionnaire pour compter la distribution des mots
dic_mots = {}
# Parcours des fichiers du corpus
for fichier in os.listdir("corpus/"):
    fh = codecs.open( os.path.join("corpus", fichier), "r", "utf-8")
    contenu = fh.read()
    mots = reg_split.split(contenu)
    nbmots += len(mots)
    fh.close()
    # On ajoute les mots à la distribution
    for mot in mots:
        if dic_mots.has_key(mot):
            dic_mots[mot] += 1
        else:
            dic_mots[mot] = 1

print "%d mots dans le corpus" % nbmots

# On extrait les 10 mots les plus présents
from operator import itemgetter
mots_sorted = sorted(dic_mots.items(), key=itemgetter(1), reverse=True)
print "Les 10 mots les plus fréquents : %s" % mots_sorted[:10]
```

Ce code est également téléchargeable à l'adresse : `http://www.fabienpoulard.info/download//Enseignements/EPUN/2011/decoupage-corpus-sarkozy.py`
Bonus : Vous pouvez exporter la distribution des mots des discours au format *mot :nb.occ.* et la soumettre au site `http://www.wordle.net/advanced`. Vous obtiendrez alors un nuage de mots dans le style de celui de la figure 1. Petit conseil : ne prenez que les cents mots les plus fréquents.

FIGURE 1 – Proposition de nuage de mots des discours de Nicolas Sarkozy.